	Type	Hits	Search Text	DBs
1	BRS	3	"null operation" and JVM	USPAT; EPO; JPO; IBM_TDB
2	BRS	6	NOP$ and JVM	USPAT; EPO; JPO; IBM_TDB
3	BRS	0	RTSJ and java	USPAT; EPO; JPO; IBM_TDB
4	BRS	0	RTSJ	USPAT; EPO; JPO; IBM_TDB
5	BRS	52	"null operation" and "instruction set"	USPAT; EPO; JPO; IBM_TDB
6	BRS	3	("null operation" and "instruction set") and (garbage near collection)	USPAT; EPO; JPO; IBM_TDB
7	BRS	0	(NOP$ or "null operation") near10 JVM	USPAT; EPO; JPO; IBM_TDB
8	BRS	271	(NOP$ or "null operation") near3 data	USPAT; EPO; JPO; IBM_TDB
9	BRS	0	((NOP$ or "null operation") near3 data) and JVM	USPAT; EPO; JPO; IBM_TDB
10	BRS	3	((NOP$ or "null operation") near3 data) and java	USPAT; EPO; JPO; IBM_TDB
11	BRS	10	(NOP$ or "null operation") and JVM	USPAT; EPO; JPO; IBM_TDB
12	BRS	7	(NOP or "null operation") and JVM	USPAT; EPO; JPO; IBM_TDB
13	BRS	5	("null operation") and JVM	USPAT; EPO; JPO; IBM_TDB
14	BRS	0	("null operation") near4 reference and java	USPAT; EPO; JPO; IBM_TDB
15	BRS	3	("null operation") near4 reference	USPAT; EPO; JPO; IBM_TDB
16	BRS	3	"null operation" and "garbage collection"	USPAT; EPO; JPO; IBM_TDB
17	BRS	10	"null operation" and java	USPAT; EPO; JPO; IBM_TDB
18	BRS	0	java near2 "null operation"	USPAT; EPO; JPO; IBM_TDB
19	BRS	0	java near10 "null operation"	USPAT; EPO; JPO; IBM_TDB

	Type	Hits	Search Text	DBs
20	BRS	0	jvm near10 "null operation"	USPAT; EPO; JPO; IBM_TDB
21	BRS	1	"instruction set" near10 "null operation"	USPAT; EPO; JPO; IBM_TDB
22	BRS	1	"null operation" near2 data	USPAT; EPO; JPO; IBM_TDB
23	BRS	4	"null operation" near2 field	USPAT; EPO; JPO; IBM_TDB
24	BRS	27	"null operation" near10 data	USPAT; EPO; JPO; IBM_TDB
25	BRS	2	("null operation" near10 data) and java	USPAT; EPO; JPO; IBM_TDB
26	BRS	0	"null operation" near10 "bit vector"	USPAT; EPO; JPO; IBM_TDB
27	BRS	3	"null operation" near10 vector	USPAT; EPO; JPO; IBM_TDB

Web Images Groups News Froogle more »

Google "null operation" Search Advanced Search Preferences

Web

Results **31** - **40** of about **4,050** for **"null operation"**. (**0.16** secon

Err Msg: "Operation not yet implemented" when scheduling to 'Excel ...
... ePortfolio/Web Desktop. Topic: Scheduling. Keywords: SCHEDULE EXCEL DATA ONLY TEXT OBJECT **NULL OPERATION** IMPLEMENTED UNIX CE85 CE. Status: Reported Issue. Product: ...
support.businessobjects.com/ library/kbase/articles/c2015225.asp - 21k - Cached - Similar pages

ILEngineer - Crummy .NET Decompiler * Copyright (C) 2001-2002 Jay ...
... Synchronize static label; int m_Goto; bool m_Block; bool m_Target; public: Statement(Blocks::Method *method, uint8_t *offset = **NULL**) : **Operation**(method, offset ...
svn.saurik.com/repos/ninetjer/ trunk/ilengineer/Statement.h - 4k - Cached - Similar pages

[PPT] Slide presentation Template
File Format: Microsoft Powerpoint 97 - View as HTML
... list). adminid bigint NOT NULL,-- Administrator's identifier. operation smallint NOT **NULL**,-- **Operation** on the container. allow tinyint ...
edg-wp2.web.cern.ch/edg-wp2/ security/voms/edg-scg-2003-07.ppt - Similar pages

[PDF] Abstract Poster
File Format: PDF/Adobe Acrobat - View as HTML
... heat and particle flux •For a balanced double null, particle control best with symmetric pumping •Designed for lower or upper single **null operation** QTYUIOP ...
202.127.205.62/2002年/学术报告/ 外10月23日/JT-60SU%20Div%20Design.pdf - Similar pages

Constants
... MPI_COMM_NULL - Null communicator MPI_OP_NULL - **Null operation** MPI_GROUP_NULL - Null group MPI_DATATYPE_NULL - Null datatype MPI_REQUEST_NULL - Null request ...
www.mpi-softtech.com/products/ cluster/mpi_pro/doc/Functions/Constants.html - 18k - Cached - Similar pages

using System; /// double implementation class CalcFunc { public ...
... static string SwitchSign(String number) { return -Double.Parse(number) + ""; public static String CalculateResult() { sign = **null**; **operation** = null; empty ...
lumumba.luc.ac.be/~kris/projects/ uiml.net/examples/CalcFunc.cs - 2k - Cached - Similar pages

using System; /// int implementation class CalcFunc { public ...
... static string SwitchSign(String number) { return -Int32.Parse(number) + ""; public static String CalculateResult() { sign = **null**; **operation** = null; empty ...
lumumba.luc.ac.be/~kris/projects/ uiml.net/examples/CalcFuncInt.cs - 2k - Cached - Similar pages

SR @ HHH - Matrix (Otaku Echoes)
... 16, Passive Alert. 21, **Null Operation**. 26, Daemon Activation. 30, Operator (Decker) Notification. 33, **Null Operation**. 37, Sparky-IC. 42, Active Alert. 47, **Null Operation**. ...
www.hoosierhackerhouse.com/mtx/ota/echo_ftoj.html - 27k - Cached - Similar pages

Re: [lug] wget question
... mypasswd@machine1/../README > > --10:05:02-- ftp://user:xxxxxxxx@machine1:21/README > > The two above are the same thing, except for the **null operation** of "cd ...
archive.lug.boulder.co.us/bymonth/2002.06/msg00033.html - 6k - Cached - Similar pages

Dynamic Query Runner

... request.getParameter("operation"); Worker worker = new Worker(); //Holds the methods to do individual operations if(operation == **null**) **operation** = "4"; //Will ...
www.imranweb.com/dynQuery.txt - 15k - Cached - Similar pages

◀ Goooooooooooogle ▶
Result Page: **Previous** 1 2 3 **4** 5 6 7 8 9 10 11 12 13 **Next**

"null operation" Search

Search within results | Language Tools | Search Tips

Google Home - Advertising Programs - Business Solutions - About Google

h g gec e ch e h e e

Search Results

Search Results for: **["null operation" and java]**
Found **3** of **129,310 searched.**

Search within Results

GO > Advanced Search > Search Help/Tip

Sort by: **Title** **Publication** **Publication Date** **Score** Binder

Results 1 - 3 of 3 **short listing**

1 Exploitng event stream interpretation in publish-subscribe systems 77
Yuanyuan Zhao , Rob Strom
Proceedings of the twentieth annual ACM symposium on Principles of distributed computing August 2001

Publish-subscribe messaging middleware typically offers limited and low-level options for quality of service, such as best-effort delivery versus reliable delivery, or ordered versus unordered. We propose a new, high-level approach to specifying quality of service, in which the consumer specifies an *event stream interpretation* function that maps an event stream into a state that represents the consumer's semantics of the stream. Under this approach, the system may deliver either the su ...

2 How things were: Programming lessons from days gone by: extreme design 77
Alan Creak
ACM SIGPLAN Notices December 2003
Volume 38 Issue 12

3 Integrating object-oriented programming and protected objects in Ada 95 77
A. J. Wellings , B. Johnson , B. Sanden , J. Kienzle , T. Wolf , S. Michell
ACM SIGAda Ada Letters June 2002
Volume XXII Issue 2

Integrating concurrent and object-oriented programming has been an active research topic since the late 1980's. There is now a plethora of methods for achieving this integration. The majority of approaches have taken a sequential object-oriented language and made it concurrent. A few approaches have taken a concurrent language and made it object-oriented. The most important of this latter class is the Ada 95 language, which is an extension to the object-based concurrent programming language Ada ...

Results 1 - 3 of 3 **short listing**

Search Results

Nothing Found

Your search for **["null operation" and jvm]** did not return any results.

You may revise it and try your search again below or click advanced search for more options.

"null operation" and jvm

SEARCH **[Advanced Search]** **[Search Help/Tips]**

Complete Search Help and Tips

The following characters have specialized meaning:

Special Characters	Description
, ( ) [	These characters end a text token.
= > < !	These characters end a text token because they signify the star of a field operator. (! is special: != ends a token.)
` @ \Q < { [!	These characters signify the start of a delimited token. These ar terminated by the end character associated with the start character.

TOP 20 WEB RESULTS out of about 354 Search took 0.498 seconds. (What's this?)

1. **Java** Advanced Imaging: Class NullDescriptor
   ... An OperationDescriptor describing the "**Null" operation**. The "**Null" operation** performs no processing ... DocURL. http://**java**.sun.com/products/**java**-media/jai/forDevelopers/jai-apidocs ...
   www.higgsolutions.com/docs/java/jai/jai-apidocs/javax/media/jai/operator/NullDescriptor.html - 30k - Cached

2. http://abone-tcs.sdl.sri.com/mirror/bro.isi.edu/JRSVP_CURRENT/ua/mgmt/applet/WebReq.ja
   ... import **java**.io.InputStream; import **java**.io.OutputStream; import **java**.io.DataInputStream; import **java**.io ... System.err.println("**Null Operation**"); douts.writeUTF("** **Null Operation** **\n ...
   abone-tcs.sdl.sri.com/mirror/bro.isi.edu/JRSVP_CURRENT/ua/mgmt/applet/WebReq.java - 23k - Cached - Mor pages from this site

3. geotools/geotools2/geotools-src/gcs-coverage/src/org/geotools/gp/Resampler.**java** - view - 1
   ... J2SE dependencies import **java**.awt.Dimension; import **java**.awt.Rectangle; import **java**.awt.RenderingHints; import **java**.awt ... initially wrapped into in a "**Null" operation**. Later, we will ...
   cvs.sourceforge.net/cgi-bin/viewcvs.cgi/geotools/geotools2/geotools-src/gcs-coverage/src/org/geotool s - 59k - Cached - More pages from this site

4. geotools/geotools2/geotools-src/gcs-coverage/src/org/geotools/gp/Resampler.**java** - view - 1
   ... J2SE dependencies import **java**.awt.Rectangle; import **java**.awt.RenderingHints; import **java**.awt.geom ... ini wrapped into in a "**Null" operation**. Later, we will change * this "Null ...
   cvs.sourceforge.net/cgi-bin/viewcvs.cgi/geotools/geotools2/geotools-src/gcs-coverage/src/org/geotool s - 59k - Cached - More pages from this site

5. Source of webl.lang.expr.SetExpr.**java**
   ... import webl.util.*; import **java**.util.*; publicclass SetExpr extends ValueExpr implements ... VersionTree vt, Ve V) { Operation R = **null; Operation** o = operations; while ...
   www.research.digital.com/SRC/WebL/sources/webl.lang.expr.SetExpr.java.html - 28k - Cached

6. http://www.ccs.neu.edu/research/demeter/DemeterJSR31/com1800/sp01/rathod/labguide/src
   options { STATIC = false; **JAVA**_UNICODE_ESCAPE = true; } PARSER_BEGIN(Parser) import **java**.util.*; imp **java**.io.*; import edu.neu.ccs.demeter.*; public class Parser { // oit is uugly.
   www.ccs.neu.edu/research/demeter/DemeterJSR31/com1800/sp01/rathod/labguide/src/gen/Parser.jj - 19k - Ca

7. http://www.cis.ksu.edu/SELab/Mak/UDGP/ProjectDoc.**java**
   ... import **java**.io.*; import **java**.awt.*; import **java**.awt.image.*; import **java**.applet.*; import ... cvTempG); } else { JOptionPane.showMessageDialog(**null, "Operation** not accessable from the ...
   www.cis.ksu.edu/SELab/Mak/UDGP/ProjectDoc.java - 32k - Cached

8. http://cs.baylor.edu/~sturgill/research/gape/gape/client/ScenarioEditor.**java**
   A domain independent ScenarioEditor/Viewer.
   cs.baylor.edu/~sturgill/research/gape/gape/client/ScenarioEditor.java - 21k - Cached

9. eNode Clients for Web Services — Fax
Table of Contents. Markup Tutorial. Interfacing with **Java**™Web Services. EMail Client. Translation Client. A Si SOAP-based Fax Client
www.enode.com/x/webservices/fax.html - 19k - Cached - More pages from this site

10. eNode Clients for Web Services — eMail
Table of Contents. Markup Tutorial. Interfacing with **Java**™Web Services. Message. Fax Client. A Simple SOA based EMail Client. Make sure you have JRE 1.3.1 or later, and **Java** Web Start 1.0.1 or later already installed, before you launch EMailer.
www.enode.com/x/webservices/email.html - 19k - Cached - More pages from this site

11. http://www.omg.org/issues/issue3250.txt
Issue 3250: Null valuetypes not supported by DynValue (orb_revision) Source: IONA (Mr. ... boolean is_null(); T is_**null operation** returns TRUE if the DynValue represents a null valuetype ... perform strict exception checkin (like **Java**) will no longer compile using the ...
www.omg.org/issues/issue3250.txt - 199k - Cached - More pages from this site

12. ModelMBeanInfo (**Java** Management Extensions 1.2 API Specification)
... **Java** resources wishing to be manageable instantiate the ModelMBean using the MBeanServer's createMBe method ... Wraps an IllegalArgumentException for a **null operation** name. ...
star.cs.byu.edu/jmx_1.2/api/javax/management/modelmbean/ModelMBeanInfo.html - 36k - Cached

13. ModelMBeanInfo (**Java** 2 Platform SE v1.5.0)
... **Java** resources wishing to be manageable instantiate the ModelMBean using the MBeanServer's createMBe method ... Wraps an IllegalArgumentException for a **null operation** name. ...
java.sun.com/j2se/1.5.0/docs/api/javax/management/modelmbean/ModelMBeanInfo.html - 41k - Cached - More pages from this site

14. http://cs.baylor.edu/~sturgill/research/gape/gape/client/OldScenarioEditor.**java**
A domain independent ScenarioEditor/Viewer.
cs.baylor.edu/~sturgill/research/gape/gape/client/OldScenarioEditor.java - 19k - Cached

15. : Class NullDescriptor
... Class NullDescriptor. **java**.lang.Object | +--javax.media.jai.OperationDescriptorImpl ... OperationDescriptor describing the "**Null" operation**. The "**Null" operation** performs no processing ...
www-mipl.jpl.nasa.gov/~rgd/OpSpec/NullDescriptor.html - 18k - Cached

16. Long (JML and MultiJava documentation)
... DETAIL: FIELD | CONSTR | METHOD. **java**.lang. Class Long. **java**.lang.Object | +-- **java**.lang ... null&&! nm.equals("")&&System.getProperty(nm) != **null&&Operation**.decodeable(System.getProperty(nm ...
www.dc.fi.udc.es/ai/tp/practica/jml/JML/javadocs/java/lang/Long.html - 53k - Cached

17. http://www.omg.org/issues/issue3135.txt
Issue 3135: valuebox and DynAny (orb_revision) Source: IONA (Mr.
www.omg.org/issues/issue3135.txt - 205k - Cached - More pages from this site

18. ModelMBeanInfoSupport (**Java** 2 Platform Ent. Ed. v1.4)
... **Java** resources wishing to be manageable instantiate the ModelMBean using the MBeanServer's createMBe method ... Wraps an IllegalArgumentException for a **null operation** name. ...
java.sun.com/j2ee/1.4/docs/api/javax/management/modelmbean/ModelMBeanInfoSupport.html - 59k - Cached More pages from this site

19. http://neem.cs.ttu.edu/jxta_src/platform_src/platform/binding/**java**/impl/src/net/jxta/impl/acces impleACL/SimpleACLAccessService.**java**

... simpleACL; import **java**.net.URL; import **java**.util.Enumeration; import **java** ... debug("Adding operation : '" + ((**null == operation**) ? "<<DEFAULT>>" : operation) + "' with " + allowed ...
neem.cs.ttu.edu/jxta_src/platform_src/platform/binding/java/impl/src/net/jxta/impl/access/simpleACL/ S - 19k - Cached

20. [Www-cvs] CVS update: reptile/src/**java**/org/openprivacy/reptile
... Modified Files: ContentTransformer.**java** ReptileServlet.**java** Log Message: login ... null) { - if (operation == **{ - operation** = "null"; + PageDispatchEngine.dispatch(request ...
mail.openprivacy.org/pipermail/www-cvs/2001-June/002035.html - 15k - Cached

Results Page:
1 2 3 4 5 6 7 8 9 10 **Next**

Help us improve your search experience. Send us feedback.

Terms used **null operation** and **JVM** Found **496** of **138,663**

Sort results by: relevance
Display results: expanded form
Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next
Best 200 shown Relevance scale

1 Executable JVM model for analytical reasoning: a study
Hanbing Liu, J Strother Moore
June 2003 **Proceedings of the 2003 workshop on Interpreters, Virtual Machines and Emulators**
Full text available: pdf(230.18 KB) Additional Information: full citation, abstract, references, index terms, review

To study the properties of the Java Virtual Machine(JVM) and Java programs, our research group has produced a series of JVM models written in a functional subset of Common Lisp. In this paper, we present our most complete JVM model from this series, namely, M6, which is derived from a careful study of the J2ME KVM [16] implementation.On the one hand, our JVM model is a conventional machine emulator. M6 models accurately almost all aspects of the KVM implementation, including the dynamic class lo ...

2 Compiling scheme to JVM bytecode:: a performance study
Bernard Paul Serpette, Manuel Serrano
September 2002 **ACM SIGPLAN Notices , Proceedings of the seventh ACM SIGPLAN international conference on Functional programming**, Volume 37 Issue 9
Full text available: pdf(298.96 KB) Additional Information: full citation, abstract, references, index terms

We have added a Java virtual machine (henceforth JVM) bytecode generator to the optimizing Scheme-to-C compiler Bigloo. We named this new compiler BiglooJVM. We have used this new compiler to evaluate how suitable the JVM bytecode is as a target for compiling strict functional languages such as Scheme. In this paper, we focus on the performance issue. We have measured the execution time of many Scheme programs when compiled to C and when compiled to JVM. We found that for each benchmark, at leas ...

Keywords: Java virtual machine, compilation, functional languages, scheme

3 Exploiting FPGA concurrency to enhance JVM performance
James Parnis, Gareth Lee
January 2004 **Proceedings of the 27th conference on Australasian computer science - Volume 26**
Full text available: pdf(221.02 KB) Additional Information: full citation, abstract, references

The Java Programming Language has been praised for its platform independence and portability, but because of its slow execution speed on a software Java Virtual Machine (JVM), some people decide to use faster languages such as C. Building a JVM in hardware is

an obvious solution to this problem. Several approaches have been taken to try to achieve the best solution. One approach is by reducing the number of Java instructions a program has to execute along with directly executing instructions in ...

Keywords: FPGA, Java virtual machine, field programmable logic

4 Java Virtual Machine: JVM versus CLR: a comparative study

Jeremy Singer

June 2003 **Proceedings of the 2nd international conference on Principles and practice of programming in Java**

Full text available: pdf(84.98 KB) Additional Information: full citation, abstract, references

We present empirical evidence to demonstrate that there is little or no difference between the Java Virtual Machine and the .NET Common Language Runtime, as regards the compilation and execution of object-oriented programs. Then we give details of a case study that proves the superiority of the Common Language Runtime as a target for imperative programming language compilers (in particular GCC).

5 A static type system for JVM access control

Tomoyuki Higuchi, Atsushi Ohori

August 2003 **ACM SIGPLAN Notices , Proceedings of the eighth ACM SIGPLAN international conference on Functional programming**, Volume 38 Issue 9

Full text available: pdf(150.01 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents a static type system for JAVA Virtual Machine (JVM) code that enforces an access control mechanism similar to the one found, for example, in a JAVA implementation. In addition to verifying type consistency of a given JVM code, the type system statically verifies that the code accesses only those resources that are granted by the prescribed access policy. The type system is proved to be sound with respect to an operational semantics that enforces access control dynamically, si ...

Keywords: JVM, access control, stack inspection, type inference, type system

6 A high performance cluster JVM presenting a pure single system image

Y. Aridor, M. Factor, A. Teperman, T. Eilam, A. Schuster

June 2000 **Proceedings of the ACM 2000 conference on Java Grande**

Full text available: pdf(916.33 KB) Additional Information: full citation, references, citings, index terms

7 JVM: platform independent vs. performance dependent

Ruben Pinilla, Marisa Gil

April 2003 **ACM SIGOPS Operating Systems Review**, Volume 37 Issue 2

Full text available: pdf(4.93 MB) Additional Information: full citation, abstract, references

Nowadays Java technology has become an important reference to application developers. The great acceptance from software developer's community is mainly based on its platform independence execution environment. In this paper, we analyze the degree of dependence between the Java Virtual Machine (JVM) and the underlying platform. We have observed that the Sun JVM (Java 2 SDK 1.2.2-006) API is independent from programmer's point of view, but offers different behaviour depending on the target HPI (H ...

Keywords: JVM, Java threads, concurrency, kernel threads, multithreaded, scheduling, user threads

8 Automatic translation of Fortran to JVM bytecode

Keith Seymour, Jack Dongarra

June 2001 **Proceedings of the 2001 joint ACM-ISCOPE conference on Java Grande**

Full text available: pdf(555.04 KB) Additional Information: full citation, abstract, references, index terms

This paper reports on the design of a FORTRAN-to-Java translator whose target language is the instruction set of the Java Virtual Machine. The goal of the translator is to generate Java implementations of legacy FORTRAN numerical codes in a consistent and reliable fashion. The benefits of directly generating bytecode are twofold. First, it provides a much more straightforward and efficient mechanism for translating FORTRAN GOTO statements. Second, it provides a framework for pursuing various ...

9 Multilanguage programming on the JVM: the Ada 95 benefits

Franco Gasperoni, Gary Dismukes

December 2000 **ACM SIGAda Ada Letters**, Volume XX Issue 4

Full text available: pdf(1.48 MB) Additional Information: full citation, abstract, index terms

The latest trend in our industry, "pervasive computing", predicts the proliferation of numerous, often invisible, computing devices embedded in consumer appliances connected to the ubiquitous Internet. Secure, reliable applications combined with simplicity of use will make or break a company's reputation in this market.The Java "write once, run anywhere" paradigm, introduced by Sun in the mid-90s, is embodied in a widely available computing platform targeting pervasive devices. Although the Java ...

10 Formalizing the safety of Java, the Java virtual machine, and Java card

Pieter H. Hartel, Luc Moreau

December 2001 **ACM Computing Surveys (CSUR)**, Volume 33 Issue 4

Full text available: pdf(442.86 KB) Additional Information: full citation, abstract, references, citings, index terms

We review the existing literature on Java safety, emphasizing formal approaches, and the impact of Java safety on small footprint devices such as smartcards. The conclusion is that although a lot of good work has been done, a more concerted effort is needed to build a coherent set of machine-readable formal models of the whole of Java and its implementation. This is a formidable task but we believe it is essential to build trust in Java safety, and thence to achieve ITSEC level 6 or Common Crite ...

Keywords: Common criteria, programming

11 A formal specification of Java class loading

Zhenyu Qian, Allen Goldberg, Alessandro Coglio

October 2000 **ACM SIGPLAN Notices , Proceedings of the 15th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 35 Issue 10

Full text available: pdf(241.45 KB) Additional Information: full citation, abstract, references, citings, index terms

The Java Virtual Machine (JVM) has a novel and powerful mechanism to support lazy, dynamic class loading according to user-definable policies. Class loading directly impacts type safety, on which the security of Java applications is based. Conceptual bugs in the loading mechanism were found in earlier versions of the JVM that lead to type violations. A deeper understanding of the class loading mechanism, through such means as formal analysis, will improve our confidence that no additional bugs a ...

12 ULT: a Java threads model for platform independent execution
Ruben Pinilla, Marisa Gil
October 2003 **ACM SIGOPS Operating Systems Review**, Volume 37 Issue 4

Full text available: pdf(3.39 MB) Additional Information: full citation, abstract, references

Java is known to be a valuable technology for building platform independent applications, based on an independent execution environment provided by a virtual machine (JVM, Java Virtual Machine) and an API formed by a set of classes. The Java platform was conceived as a solution for application transportation between heterogeneous platforms without the need of adapting and recompiling the source code. Some previous analyses of Sun JVM implementation (Java 2 SDK 1.2.2-006) establish that the HPI (...

Keywords: HPI, JVM, Java threads, ULT, concurrency, kernel threads, multithreaded, scheduling, user threads

13 Persistent execution state of a Java virtual machine
Takashi Suezawa
June 2000 **Proceedings of the ACM 2000 conference on Java Grande**

Full text available: pdf(709.96 KB) Additional Information: full citation, references, citings, index terms

Keywords: JVM, Java, checkpointing, execution state, persistence, recovery

14 Providing soft real-time QoS guarantees for Java threads
James C. Pang, Gholamali C. Shoja, Eric G. Manning
June 2001 **Proceedings of the 2001 joint ACM-ISCOPE conference on Java Grande**

Full text available: pdf(853.96 KB) Additional Information: full citation, abstract, references, index terms

The Java platform has many characteristics that make it very desirable for integrated continuous media processing. Unfortunately, it lacks the necessary CPU resource management facility to support quality of service guarantees for soft real-time multimedia tasks. In this paper, we present our new Java Virtual Machine, Q-JVM, which brings CPU resource management to the Java platform. Q-JVM is based on Sun's JVM version 1.1.5. It implements an enhanced version or the MTR-LS algorithm in its thr ...

15 Techniques for obtaining high performance in Java programs
Iffat H. Kazi, Howard H. Chen, Berdenia Stanley, David J. Lilja
September 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 3

Full text available: pdf(816.13 KB) Additional Information: full citation, abstract, references, citings, index terms

This survey describes research directions in techniques to improve the performance of programs written in the Java programming language. The standard technique for Java execution is interpretation, which provides for extensive portability of programs. A Java interpreter dynamically executes Java bytecodes, which comprise the instruction set of the Java Virtual Machine (JVM). Execution time performance of Java programs can be improved through compilation, possibly at the expense of portabili ...

Keywords: Java, Java virtual machine, bytecode-to-source translators, direct compilers, dynamic compilation, interpreters, just-in-time compilers

16 Multitasking without comprimise: a virtual machine evolution

Grzegorz Czajkowski, Laurent Daynés
October 2001 **ACM SIGPLAN Notices , Proceedings of the 16th ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications**, Volume 36 Issue 11

Full text available: pdf(220.97 KB) Additional Information: full citation, abstract, references, citings, index terms

The multitasking virtual machine (called from now on simply MVM) is a modification of the Java virtual machine. It enables safe, secure, and scalable multitasking. Safety is achieved by strict isolation of application from one another. Resource control augment security by preventing some denial-of-service attacks. Improved scalability results from an aggressive application of the main design principle of MVM: share as much of the runtime as possible among applications and replicate everything el ...

Keywords: Java virtual machine, application isolation, native code execution, resource control

17 Targeting GNAT to the Java virtual machine
Cyrille Comar, Gary Dismukes, Franco Gasperoni
November 1997 **Proceedings of the conference on TRI-Ada '97**

Full text available: pdf(1.72 MB) Additional Information: full citation, references, citings, index terms

18 Java bytecode as a typed term calculus
Tomoyuki Higuchi, Atsushi Ohori
October 2002 **Proceedings of the 4th ACM SIGPLAN international conference on Principles and practice of declarative programming**

Full text available: pdf(214.36 KB) Additional Information: full citation, abstract, references, citings, index terms

We propose a type system for the Java bytecode language, prove the type soundness, and develop a type inference algorithm. In contrast to the existing proposals, our type system yields a typed term calculus similar to type systems of lambda calculi. This enables us to transfer existing techniques and results of type theory to a JVM-style bytecode language. We show that ML-style let polymorphism and recursive types can be used to type JVM subroutines, and that there is an ML-style type inference ...

Keywords: Java bytecode, bytecode verifier, type inference, type system

19 Implementing jalapeño in Java
Bowen Alpern, C. R. Attanasio, Anthony Cocchi, Derek Lieber, Stephen Smith, Ton Ngo, John J. Barton, Susan Flynn Hummel, Janice C. Sheperd, Mark Mergen
October 1999 **ACM SIGPLAN Notices , Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 34 Issue 10

Full text available: pdf(1.57 MB) Additional Information: full citation, abstract, references, citings, index terms

Jalapeño is a virtual machine for Java™ servers written in Java.A running Java program involves four layers of functionality: the user code, the virtual-machine, the operating system, and the hardware. By drawing the Java / non-Java boundary below the virtual machine rather than above it, Jalapeño reduces the boundary-crossing overhead and opens up more opportunities for optimization.To get Jalapeño started, a boot image of a ...

20 A compositional account of the Java virtual machine

Phillip M. Yelland

January 1999 **Proceedings of the 26th ACM SIGPLAN-SIGACT symposium on Principles of programming languages**

Keywords: Haskell, Java bytecode, Java virtual machine, verification

h c g e cf c

javax.media.jai.operator

Class NullDescriptor

```
java.lang.Object
  |
  +--javax.media.jai.OperationDescriptorImpl
        |
        +--javax.media.jai.operator.NullDescriptor
```

All Implemented Interfaces:
OperationDescriptor, RegistryElementDescriptor, Serializable

---

public class **NullDescriptor**
extends OperationDescriptorImpl

An `OperationDescriptor` describing the "Null" operation.

The "Null" operation performs no processing. It merely propagates its first source along the operation chain unmodified. There may be an arbitrary number of sources but only the first one is passed along so it must hav the appropriate class type for the operation mode.

This operation may be useful as a placeholder in operation chains and in creating nodes to which `PropertyGenerators` may be attached. This would enable non-image data nodes to be present in chains without requiring that specific `OperationDescriptors` be implemented for these operations. The `PropertyGenerators` required would in this case be added locally to the nodes using the `addPropertyGenerator()` method of the node.

Resource List

Name	Value
GlobalName	Null
LocalName	Null
Vendor	com.sun.media.jai
Description	An operation which does no processing.
DocURL	http://java.sun.com/products/java-media/jai/forDevelopers/jai-apidocs/javax/media/jai/operator/NullDescriptor.html
Version	1.0

No parameters are needed for this operation.

Since:

h c c e f e e e c e e

JAI 1.1

See Also:

OperationDescriptor, Serialized Form

Fields inherited from class javax.media.jai.OperationDescriptorImpl
resources, sourceNames, supportedModes

Fields inherited from interface javax.media.jai.OperationDescriptor
NO_PARAMETER_DEFAULT

Constructor Summary

NullDescriptor() Constructor.

Method Summary

static RenderedOp	**create**(RenderedImage source0, RenderingHints hints) An operation which does no processing.
static RenderableOp	**createRenderable**(RenderableImage source0, RenderingHints hints) An operation which does no processing.
Object	**getInvalidRegion**(String modeName, ParameterBlock oldParamBlock, RenderingHints oldHints, ParameterBlock newParamBlock, RenderingHints newHints, OperationNode node) Calculates the region over which two distinct renderings of the "Null" operation m be expected to differ.
protected boolean	**validateSources**(String modeName, ParameterBlock args, StringBuffer msg) Returns true if there is at least one source and the first source is a RenderedImage or RenderableImage.

Methods inherited from class javax.media.jai.OperationDescriptorImpl
arePropertiesSupported, getDefaultSourceClass, getDestClass, getDestClass, getName, getNumParameters, getNumSources, getParamClasses, getParamDefaults, getParamDefaultValue, getParameterListDescriptor, getParamMaxValue, getParamMinValue, getParamNames, getPropertyGenerators, getPropertyGenerators, getRenderableDestClass, getRenderableSourceClasses, getResourceBundle, getResources, getSourceClasses, getSourceClasses, getSourceNames, getSupportedModes, isImmediate, isModeSupported, isRenderableSupported, isRenderedSupported, makeDefaultSourceClassList, validateArguments, validateArguments, validateParameters, validateParameters, validateRenderableArguments, validateRenderableSources, validateSources

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

h c c e f e e e c e e

Constructor Detail

NullDescriptor

```
public NullDescriptor()
```

Constructor.

Method Detail

validateSources

```
protected boolean validateSources(String modeName,
                                  ParameterBlock args,
                                  StringBuffer msg)
```

Returns `true` if there is at least one source and the first source is a `RenderedImage` or `RenderableIma`

Overrides:

`validateSources` in class `OperationDescriptorImpl`

Throws:

`IllegalArgumentException` - if `args` is `null`.

`IllegalArgumentException` - if `msg` is `null` and the validation fails.

---

getInvalidRegion

```
public Object getInvalidRegion(String modeName,
                               ParameterBlock oldParamBlock,
                               RenderingHints oldHints,
                               ParameterBlock newParamBlock,
                               RenderingHints newHints,
                               OperationNode node)
```

Calculates the region over which two distinct renderings of the "Null" operation may be expected to differ.

The operation returns an empty `Shape` if the first source in each of the two `ParameterBlock`s are equa according to the `equals()` method of the old source or `null` for all other cases.

Overrides:

`getInvalidRegion` in class `OperationDescriptorImpl`

Parameters:

`modeName` - The name of the mode.

`oldParamBlock` - The previous sources and parameters.

`oldHints` - The previous hints.

`newParamBlock` - The current sources and parameters.

h c c e f e e e c e e

newHints - The current hints.
node - The affected node in the processing chain (ignored).

Returns:

The region over which the data of two renderings of this operation may be expected to be invalid null if there is no common region of validity. A non-null empty region indicates that the operation would produce identical data over the bounds of the old rendering although perhaps no over the area occupied by the *tiles* of the old rendering.

Throws:

IllegalArgumentException - if modeName is null or if either oldParamBlock or newParamBlo is null.
IllegalArgumentException - if oldParamBlock or newParamBlock does not contain at least o source.

---

create

```
public static RenderedOp create(RenderedImage source0,
                                RenderingHints hints)
```

An operation which does no processing.

Creates a ParameterBlockJAI from all supplied arguments except hints and invokes JAI.create (String,ParameterBlock,RenderingHints).

Parameters:

source0 - RenderedImage source 0.
hints - The RenderingHints to use. May be null.

Returns:

The RenderedOp destination.

Throws:

IllegalArgumentException - if source0 is null.

See Also:

JAI, ParameterBlockJAI, RenderedOp

---

createRenderable

```
public static RenderableOp createRenderable(RenderableImage source0,
                                            RenderingHints hints)
```

An operation which does no processing.

Creates a ParameterBlockJAI from all supplied arguments except hints and invokes JAI.createRenderable(String,ParameterBlock,RenderingHints).

Parameters:

source0 - RenderableImage source 0.
hints - The RenderingHints to use. May be null.

h c c e f e e e c e e

Returns:

The `RenderableOp` destination.

Throws:

`IllegalArgumentException` - if `source0` is `null`.

See Also:

`JAI`, `ParameterBlockJAI`, `RenderableOp`

Overview Package Class Tree Deprecated Index Help

PREV CLASS NEXT CLASS FRAMES NO FRAMES

SUMMARY: INNER | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD

---

h c c e f e e e c e e

protected boolean	**validateRenderableSources**(ParameterBlock args, StringBuffer msg) **Deprecated.** *as of JAI 1.1 in favor of* `validateSources("renderable", ...`
protected boolean	**validateSources**(ParameterBlock args, StringBuffer msg) **Deprecated.** *as of JAI 1.1 in favor of* `validateSources("rendered", ...)`
protected boolean	**validateSources**(String modeName, ParameterBlock args, StringBuffer msg) Returns `true` if this operation supports the specified mode, and is capable of handling the given input source(s) for the specified mode.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

resources

```
protected final String[][] resources
```

The resource tags and their corresponding data, stored as an two-dimensional `String` array.

Since:

JAI 1.1

---

supportedModes

```
protected final String[] supportedModes
```

An array of operation modes supported by this operator. Must be a non-empty subset of "rendered", "renderable", "collection" and "renderableCollection" or other image operator modes to be defined late

Since:

JAI 1.1

---

sourceNames

```
protected final String[] sourceNames
```

An array of `Strings` that are the names of the sources of this operation. The names must be listed in th order corresponding to the source `Classes`.

Since:

JAI 1.1

h c c e f e e e c e e e c